Connecting XForms to Databases
An Extension to the XForms Markup Language

Mikko Honkala, Oskari Koskimies  Markku Laine
Nokia Research Center        Helsinki University Of Technology

W3C Workshop on Declarative Models of Distributed Web Applications,
5 – 6 June 2007, Dublin, Ireland
Outline of this presentation

Introduction and Problem Statement

The XFormsDB Language

Implementation

Conclusions
Introduction and Problem Statement
Introduction

- Authoring even simple multi-user Web applications is a complex task
- The author has to master many programming languages and paradigms
  - Client: HTML+JavaScript+CSS
  - Server: PHP / RoR / J2EE / etc.
  - DB: SQL / Object-relational mapping
- The success of spreadsheets has shown that there is a large number of potential non-programmer service authors
Problem Statement

- Traditionally server applications are authored like this:

- It is not enough to write the form(s) to make an application
  - For each form, there has to be a servlet which provides the binding between form and database
  - In other words, we are required to deploy application-specific proxies – not a good idea if you want to make applications easily
- Business logic is divided between form and servlet
  - Different programming paradigms
- Tricky concurrent multi-user support
A Simpler Way:
“Single-Document Service Authoring”

- What if the form itself contained the bindings to the database, and authoring that form would be enough to create a service?
Proposed solution: XFormsDB
**XFormsDB**

- Is a pure superset of XForms 1.1 + CSS 2.1
- Extends XForms with database functionality
  - Database joins, transactions, multiuser support with synchronizations, database error handling, etc.
- XPath is a too limited a query language, XQuery is used
  - SELECT can use XQuery and create new structures (e.g., joins)
  - UPDATE uses XPath to select and update an XML fragment
- A design criterion: Implementation MUST NOT require an XForms client (transformation into XHTML+CSS+JS)
**XFormsDB: Language Extensions for DB Access**

- **xformsdb:instance**
  - Extends xforms:instance
  - Top-level placeholder for a query and parameters

- **xformsdb:query**
  - The actual query (referenced or included)

- **xformsdb:submission**
  - Extends xforms:submission
  - @expressiontype
    - “select” / “update”
  - @queryinstance
    - IDREF to the xformsdb:instance element

- **Event xformsdb-query-error**
Example : SELECT

```xml
<xformsdb:instance id="select-queryinstance">
  <query xmlns="" datasrc="exist"
         doc="helloworld.xml">
    <expression>
      for $helloworld in
          /root/helloworld/message
      return $helloworld
    </expression>
  </query>
</xformsdb:instance>
<xforms:instance id="instance"/>
<xformsdb:submission id="select-querysubmission" replace="instance"
                      instance="instance" queryinstance="select-queryinstance"
                      expressiontype="select" />
```
Example: UPDATE

```xml
<xformsdb:instance id="update-queryinstance">
  <query xmlns="" datasource="exist" doc="helloworld.xml">
    <expression>
      /root/helloworld/message
    </expression>
  </query>
</xformsdb:instance>
<xforms:instance id="updinstance"/>
<xformsdb:submission id="select4update-querysubmission" replace="updinstance"
  instance="instance" queryinstance="update-queryinstance"
  expressiontype="select">
</xformsdb:submission>
<xformsdb:submission id="update-querysubmission" replace="updinstance"
  instance="instance" queryinstance="update-queryinstance"
  expressiontype="update">
</xformsdb:submission>
```
Synchronization

• The language supports multi-user applications
• Several users can make UPDATEs simultaneously
• The framework does automatic 3-way synchronization of the updated XML fragments
  • The edited fragment
  • The original fragment
  • The current DB state
• If the synchronization fails (e.g., merge conflict), it is reported to the form, which can handle the error in case-by-case basis
Future work: Transactions

• Synchronization does not solve traditional transactional problems
  • e.g. moving money from one account to another
• We can implement easy-to-use transactions by simple grouping of fragment updates: If there is a merge conflict for any member of the group, all updates fail
  • In the form the grouping can be expressed e.g. simply by placing the elements representing the fragments inside a grouping tag
• For very complex transactions, explicit transaction control similar to current database access APIs might be needed
Implementation
(1) GET XFormsDB File => XHTML+ XForms 1.1

Web Browser Supporting XForms 1.1

(2) POST XML Query Instance Data => XML Result

Generic Servlet

Request Handler

Response Handler

Server-Side XForms Processor (Orbeon)

HTML+ CSS + JavaScript

XHTML+ XForms 1.1

Presentation Tier

XFormsDB File

XFormsDB Transformer

Application Server Tier

Active XML Results

Active Queries

Query Manager

XQuery / XPath

XML Result

Integration Service Tier

Data Access Objects

Middleware

SQL Translator

OQL Translator

Virtual XML View

XML Database

XML Files

Data Tier

Relational Database

Object Database
Use Case: Blog Editor
Sorry, no posts matched your criteria.

<table>
<thead>
<tr>
<th>ARCHIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ June 2007</td>
</tr>
<tr>
<td>▪ May 2007</td>
</tr>
<tr>
<td>▪ April 2007</td>
</tr>
<tr>
<td>▪ September 2006</td>
</tr>
<tr>
<td>▪ August 2006</td>
</tr>
</tbody>
</table>
W3C Workshop on Declarative Models of Distributed Web Applications

MAREKU LAINE // June 1, 2007

Notes and comments related to the W3C workshop held in Dublin, Ireland.

Comments (1)

NIKKO HUNKALA // June 1, 2007

Remember to book flights.

LEAVE A COMMENT

Name:

Submit
ARCHIVE FOR APRIL 2007

**Phone Book -demosovellus**
MARKU LAINE // April 15, 2007
Ajatukset/muistinpanoja/havaintoja Phone Book -demosovelluksesta.
Comments (0)

**Blog-demosovellus**
MARKU LAINE // April 15, 2007
Ajatukset/muistinpanoja/havaintoja Blog-demosovelluksesta.
Comments (2)

ARCHIVE

- June 2007
- May 2007
- April 2007
- September 2006
- August 2006
WRITE A POST

Markku Leine

W3C Workshop on Declarative Mod

Headline:

Notes and comments related to the W3C workshop held in Dublin, Ireland.

Submit

ACTIONS

Write a post

Manage posts

Manage comments

Copyright © XFormsDB 2007. All rights reserved. | Public
Conclusions
Main Benefits

- Web applications can be authored using a single document and a single paradigm
  - Simultaneous multi-user access is built in to the framework
  - Easy to understand, things that are logically related are close-by
  - Simple applications are simple to write – in the cases where application data fits in client memory, writing an entire web application is no more complex than writing a simple XForms form
  - Quick development turn-around time
  - Deployment to AJAX web browsers

- Well suited for mobile:
  - XForms is more device independent than HTML forms – works better with mobile clients
Secondary Benefits

• Storing application data as an XML document makes extending simple applications easier than with relational databases
  • Applications can be written incrementally without having to worry about database schemas

• Compared to approaches which send database queries (such as XQuery or SQL) from client to database (e.g. over SOAP), our Query-ID based solution is more secure
  • A malicious client cannot modify the query and does not even know what it looks like

• Grouping of fragments into all-or-nothing updates provides an intuitive transactional model
Related Work

• End-to-End Web authoring systems which can generate forms based on database schemas

• [http://www.w3c.rl.ac.uk/pasteevents/XML_Access_Languages/Mark/xforms-xquery.html](http://www.w3c.rl.ac.uk/pasteevents/XML_Access_Languages/Mark/xforms-xquery.html)

• [http://www.rpbouret.com/xml/XMLAndDatabases.htm](http://www.rpbouret.com/xml/XMLAndDatabases.htm)


• [http://internet-apps.blogspot.com/search/label/exist](http://internet-apps.blogspot.com/search/label/exist)
Thank you!

Questions? Comments?